Testing Challenges in Microsoft Product Teams

Grant George

Office Test

Test topics

- **Context that's useful to know**
  - Product cycle overview
  - Technology / Product Shifts
  - What if…
- **Knowledge / capturing state in the product cycle**
  - Key concerns today
  - Test challenges / thought fodder

Context that's useful to know

- 1:1 dev to test ratio
- Never enough automation
- Masked costs through temp use
- DLL hell
- Growing config coverage matrix

Office Product Cycle Overview

- Vision phase
  - Test drives past-release post-mortem and
  - ..finishes past release skus
  - ..scopes new features
  - ..specs/develops new tools
- Dev milestones in new release
  - Earliest feature testing and spec inspections
  - Review, re-prioritize auto-test arsenal
  - Develop test plans, test design specs, test schedule, round up dependencies and delivery plan, loc plans, finalize tools
  - Test M features as deeply as code health allows

Office Product Cycle Overview

- Code Complete - Stabilization
  - Intense feature testing
  - Complete new automation
  - Expand test scope (intl, config, interop, loc)
- Beta
  - Further feature testing, stabilization
  - Push automation, coverage bar to goals
  - Prioritize/resolve beta, dogfood input
- Ship
  - Final scenario, top-down testing, triage, change control
  - Push to rtm

Internal shifts

- Shared code
- Worldwide exe
- LPK
- Temp versus full-time test resources
- Lack of sameness across sku-bound teams
- More cross-team feature integration
- DLL hell

External shifts

- Market success
- Customer satisfaction focus
- Extended product support life
- Institutionalized sustaining engineering

What if …

- Developer makes, synchs, private builds code change
- Prior to checkin identifies
  - potential perf impact
  - auto tests available to run (pre-flight)
  - loc impact
  - test area owners impacted by change
- During build
  - Build, CIT, BVT logs richer, auto-log, notification mechanism, etc
- Component setup requirements, state connected

What if …

- Loc build production completely automated
- Highest priority auto test collateral auto-executed
- Change impact captured, channeled correctly/quickly
- IV data channeled into optimization scenarios
- Build tools, Raid, source control as KM solution

What Matters ?

- Tightest possible loop between dev change and test impact
- Fastest possible build turn-around/throughput
- Highest reasonable barriers to breaking changes
  - The cheapest bug is the one found earliest or prevented in the first place
- Highest level of stability throughout product cycle
- Toolset that allows better automation and distribution of test burden

Knowledge in the product cycle

- We have state of the art tools and approach, but
- Gaps in current technologies – no *real* DNS / KM
- Poor integration between systems
  - for automated test creation/management/distribution with
  - manual test case management solutions with
  - defect tracking system with
  - customer (beta and post-ship) feedback mechanisms with

    code change and build management

    articulate and capture feature validation scenarios
- Product group schema needed, difficult to attain

Test challenges & thought fodder

- Cultural: Art versus Science / manufacturing
  - Recent allowances – checkin, ui automation, perf markers
- Knowledge management (driven by test)
  - [redacted] - cool and state of the art but not well-integrated, time-consuming to maintain, multiple owners
  - Product parts and owners –a survey, hard to keep current/complete
  - Legacy support exacerbated by need to re-discover old decisions support old methods
- Sub-par tools and ownership

  Portal correctness: not universal / all locales

  Loc tools (improving)

  Security (improving)

Test challenges & thought fodder

- Humans fill the gaps
  - Lack of better tool integration, change impact knowledge, workable "dns" requires human glue
  - Often fails to testing (end-game)
  - Human glue takes away from otherwise test time (code reviews, white box testing, raw bug discovery)
- Mixed history of tools delivery
  - Rational Rego - good

    Other key tools home-grown

    Cross-team tool disconnect

Test challenges & thought fodder

- Build process
  - sig. improved in Office 10, but not fast nor stable enough yet
  - company-wide issue
- Better automation needed for
  - various html versions to support
  - staying backward compatible
  - browser testing (as host)
- Incorporating prioritized test matrix into execution methods
- Easily instantiatable real-world test configurations
  - Post-ship bug trends suggest more emphasis
  - Current drive-snapshot tools don't cut it

Test challenges & thought fodder

- **Cross-team test liabilities**
  - Who ships after us owns keeping us working
    - doesn't work anymore
    - Insufficient experience
    - automation not completely canned - still needs area-expertise
  - Exporting test collateral to other teams – difficult at best

How can MS Research help ?

- Product development cycle is very complex
- A few good examples out of which the best could be culled
- Key people sit with product teams esp dev and test
- Understand how to catch and prevent error earlier
- Capture and manage more comprehensive "state"

Questions ?